Annex I – CONCEPTUAL MODEL EXAMPLES

This document deliberately presented guidance, as opposed to specification, to conceptual modeling to capture the essential best-practices while remaining tailorable to a broad range of enterprise contexts. In this line, this annex presents examples to illustrate the various Process Activities and Products.

The objective of this annex is to guide the reader in the implementation of a conceptual modeling process and conceptual model products in its own enterprise context while avoiding restricting it to standard templates. The examples are intended to clarify the most abstract parts of the guidance, to bring out the range of applicability and to expose important issues.

For conciseness and time constraints, no thorough end-to-end example is included and trivial parts have been omitted. The examples have been selected amongst a number of current enterprise practices. The domain covered by the examples must not be taken as a limitation to the scope of applicability of the guidance.

Example I-1 differentiates between conceptual model space, mission space and simulation space requirements and demonstrates how to derive knowledge needs from requirements. Example I-2 presents a method to: gather, structure and document knowledge; generate domain ontology; and, use this knowledge to design and build conceptual model artefacts. Example I-3 differentiates the representation of mission-space knowledge, simulation-space knowledge and the conceptual model of a simulation. Example I-4 presents sample conceptual model artefacts based on a community-specific conceptual model design. Finally, Example I-5 illustrates the iterative evolution of a conceptual model requirements, design and artefacts.

I.1 DEFINING CONCEPTUAL MODEL REQUIREMENTS AND DERIVING KNOWLEDGE NEEDS

I.1.1 Process Activity 2.1 – Identify, Analyze and Record Conceptual Model, Mission and Simulation Space Requirements

In Process Activity 2.1, the conceptual model requirements are identified, analyzed and recorded. It is suggested to address the conceptual model requirement definition in terms of conceptual model space, mission space and simulation space requirements. The objective of this example is to differentiate between conceptual model space, mission space and simulation space requirements and to demonstrate what is inclusive in the conceptual modeling process. This example was developed by the MSG-058 Task Group in testing several requirements against the three-space classification. Although very partial, the artefact in Table I-1 is an example of Product 2.1 – Conceptual Model Requirement Specification. The conceptual model requirement classification may be arbitrary and is not error proof. The key point is to be all inclusive when capturing conceptual model requirements.

Table I-1: Examples of Conceptual Model Requirements Categorized in Terms of Conceptual Model Space, Mission Space and Simulation Space.

Conceptual Model Space **Conceptual Model** requires…	**Mission Space** **Conceptual Model** requires…	**Simulation Space** **Conceptual Model** requires…
To have views adapted to each stakeholder	*To represent a battalion*	*To represent a decision-maker training simulation*
To be useful for interoperability within NATO	*To represent the command and control system*	*To represent the live usage of a decision support tool*
To be readable by a computer	*To represent a peace keeping mission*	*To represent fair fight*
To be readable in French	*To represent insurgents*	*To represent an HLA simulation*

I.1.2 Process Activity 2.4 – Derive Mission Space and Simulation Space Knowledge Needs

In Process Activity 2.4, the mission space and simulation space knowledge needs are derived. The objective of this example is to demonstrate how to derive knowledge needs from requirements. Table I-2 presents knowledge needs derived from a few requirements of Table I-1. Although very partial, the artefact in Table I-2 is an example of Product 2.2 – Conceptual Model Knowledge Acquisition Needs. Deriving knowledge needs requires experience and it is more easily done iteratively. The outcome can turn out to be arbitrary, thus a special effort must be made to avoid preconceived ideas to produce an objective knowledge need statement.

Table I-2: Examples of Conceptual Model Knowledge Needs Derived from Some of the Sample Conceptual Model Requirements of Table I-1.

Space	**Conceptual Model Requirements**	**Conceptual Model Knowledge Needs**
Simulation Space + Mission Space	• *To represent a battalion* • *To represent the live usage of a decision support tool*	• *Need knowledge on battalion composition, humans, equipments, etc., at the level of detail supported by the decision tool and at the level of fidelity detectable by the tool* • *Need knowledge from a tank driver, from a physicist, etc.*
Simulation	• *To represent the live usage of a decision support tool*	• *Need knowledge in the decision support tool inputs* • *Need the minimum detectable thresholds for each input*
Simulation	• *To represent an HLA simulation*	• *Need knowledge on the HLA concepts: classes, interactions, time management, data management, etc.*
Simulation	• *To represent a decision-maker training simulation*	• *Need knowledge on the human interface* • *Need knowledge on the evaluation metrics*

I.2 FROM KNOWLEDGE ACQUISITION TO KNOWLEDGE USE IN A CONCEPTUAL MODEL

This example is taken from the Swedish Defence Conceptual Modeling Framework (DCMF) Project [1]. The DCMF process could be an implementation of the proposed conceptual modeling guidance. It is divided in four main phases: knowledge acquisition, knowledge representation, knowledge modeling and knowledge use.

The example is written as straight forward steps to introduce sample product artefacts although, in reality, the knowledge acquisition and the conceptual model construction have been done iteratively. The proposed conceptual modelling guidance does not prevent to produce conceptual model artefacts to represent the knowledge as it is acquired to help acquiring more knowledge. In fact, it is rather advised to do so. This is part of the modeling art.

This example presents the conceptual model of a scenario, as opposed to the conceptual model of a system. The sample scenario is taking place in Kosovo and its surroundings, in May 2002. NATO forces are conducting a Peace Support Operation in order to regain stability and security in Kosovo. A Swedish patrol (KS05) from the Swedish peace keeping force discovers a looted weapons depot and report this into the information system of the Swedish Intelligence. An intelligence officer in Sweden receives the report and starts a further investigation. Information from different sources leads to the estimate that the missing weapons might be smuggled to Sweden by organized criminals. Cooperation between different military and civil organizations to acquire information leads to the confiscation of the weapons in the harbor of Gothenburg in Sweden.

I.2.1 Process Activity 3.4 – Gather, Structure and Document Knowledge

Process Activity 3.4 consists in gathering, structuring and documenting knowledge to ultimately produce a conceptual model. The objective of this example is to illustrate a method for performing Process Activity 3.4.

In this example, knowledge acquisition was performed using video clippings and in-depth interviews carried out with subject-matter experts for further clarification and enrichment of the scenario description. It resulted in a description of the scenario in natural language, for which an excerpt corresponding to paragraph 1 is presented in Table I-3.

Table I-3: Sample Scenario Description in Natural Language (Paragraph 1).

A Swedish patrol from a battalion in Kosovo finds weapons in the forest near a village called Janjevo. The finding is reported in the battalion's intelligence report and this is transferred in code to Stockholm. The information about the finding is received by the Intelligence Division at MUST and the report is registered in the System. The information about the found weapons is made available for the department of international intelligence (MUST IntUnd).

Then, implicit and explicit knowledge was represented from the natural language description. Table I-4 presents some implicit knowledge inferred from paragraph 1 of the scenario description in natural language.

Table I-4: Sample Implicit Knowledge Inferred from the Scenario Description.

- *There is a Peace Support Operation in Kosovo sometime in MAY 2002, of which the Swedish contingent is a part of. (Inferred from background context material).*
- *Janjevo is a geographical location in Kosovo.*
- *There is a forest near Janjevo.*
- *Swedish troops go on regular patrol missions.*
- *There is a procedure (military) to be followed by any military PATROL if they are on a patrolling mission. It also implies that there would be standard operating procedures and regulations governing this process of patrolling.*
- *'Swedish' implies that Sweden is a sovereign nation, and that it has military capability, and is part of the UN Peace Support Operations.*
- *Weapons are hidden, that is, they are obscured from normal sight and they are not left for public viewing.*

Explicit knowledge can be extracted using different methods. Table I-5 presents the results of the Five Ws method applied on paragraph 1 of the scenario description in natural language.

Table I-5: Sample Explicit Knowledge Extracted from the Scenario Description Using the Five Ws Method.

Who	Patrol from Swedish contingent KS05	Patrol: Military type Organization (under govt. type object-type: Unit-Type: has Affiliation object type relation to Swedish Contingent: Object-Item-Group Swedish: Affiliation
What	Patrolling	Patrolling: Action-task purpose: WHY AOI: Location
When	From 1st May to 31st May 2002	
Why	To secure AOI	
Where	AOI somewhere in Kosovo	AOI: Specification detail AOI is both a Location as well as a CONTROL FEATURE: may even be a sub-type of control – feature like ROUTE, etc.

Table I-6 presents the results of the Knowledge Meta Meta Model (KM3) methodology [1] applied on paragraph 1 of the scenario description in natural language.

Table I-6: Sample Explicit Knowledge Extracted from the Scenario Description Using the KM3 Method.

Entity Type :: Swedish patrol
Entity Type :: Contingent in Kosovo
ElementComposition : <Swedish patrol , Contingent in Kosovo>
Entity Type :: Weapons
Entity Type :: Forest
Attribute : Close to ET:Janjevo, in kilometers
    CompositionType : RangedAttribute
    Domain : Distance
    StartValue : 0
    StopValue : 10
Entity Type :: Janjevo
Attribute : Village in Kosovo
    CompositionType : RangedAttribute
    Domain : Inhabitants
    StartValue : 100
    StopValue : 1000
Action Type :: Finds
Time : May 2002
RoleInAction : <finder, patrol>
RoleInOrganisationType : <patrol, ET: Swedish patrol>
Criterion :: SF: (prob 1, isStartCriterion t,
[Swedish patrol : onPatrol AND Forest AND Weapons])
State: found weapons
ActivityState : Finding weapons (has occurred) / activity Finds has occurred */*

Entity Type : Swedish patrol
    State: alerted AND onPatrol
Entity Type :: Weapons
    State: found

I.2.2 Process Activity 3.5 – Generate/Extend a Domain Ontology

In Process Activity 3.5, the knowledge captured in previous activities is formalized semantically as domain ontology. The objective of this example is to illustrate how to generate domain ontology. This example uses the knowledge of example I-2.1 to translate it in domain ontology.

The knowledge was modeled through a semantic mapping of the semi-structured information. The implicit knowledge was merged with the explicit knowledge in a machine readable format, namely in the DCMF ontology developed using the Web Ontology Language (OWL) [2]. Table I-7 lists some of the concept (class) types that were instantiated to model the sample scenario paragraph.

Table I-7: Sample Ontology Classes Instantiated to Document the Scenario.

- *Action-Task*
- *Action-Task-Status*
- *Action-Objective-Type (securing area of interest)*
- *Action-Required-Capability (for patrol mission) related to*
- *Action-Event-Status: (through this it is associated to action-event. Finding weapons in a particular action-task: patrolling.)*
- *Reporting-data*
- *Action-Event*
- *Object-Item-Group-Account: (the composition or relation of object types involved in the patrolling action.)*
- *Capability: sub type: Mission-Capability: (specifies required parameters for carrying out a patrol.)*
- *Affiliation*
- *Context*
- *Location*
- *Control Feature*
- *Action-Temporal-Association: time events, sequences and info for placing the action tasks and events in temporal sequence*
- *Object-Type:Equipment-Type:Non-Consummable-Equipment-Type:Weapons*

Table I-8 presents an excerpt of the OWL code format for the Patrol Mission instance of the Action-Required-Capability concept from the sample scenario paragraph. The artefact in Table I-8 is an example of Product 3.1 – Validated Knowledge.

Table I-8: Sample OWL Code for the Patrol Mission Instance of the Action-Required-Capability Class Capturing the Scenario Knowledge.

```
<Action-Required-Capability rdf:ID="patrolreqdqty">
 <quantifies>
 <Mission-capability rdf:ID="patrolmission">
 <capability-id rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >CMmscapability2</capability-id>
 <capability-unit-of-measure-code rdf:datatype= "http://www.w3.org/2001/XMLSchema#string"
 >square-metres-per-hour</capability-unit-of-measure-code>
 <is-quantified-in rdf:resource="#patrolreqdqty"/>
 <capability-subcategory-code rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >maximum-Range</capability-subcategory-code>
 <capability-category-code rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >military-load-capability</capability-category-code>
 <capability-day-night-code rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >day-and-night</capability-day-night-code>
 </Mission-capability>
 </quantifies>
 <capability-id rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >CMmscapabilityid1</capability-id>
 <is-minimum-required-for rdf:resource="#actioneventstatus1"/>
 <action-required-capability-quantity rdf:datatype=

 "http://www.w3.org/2001/XMLSchema#int">15</action-required-capability-quantity>
 <action-id rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >patrol_reqd_qty</action-id>

</Action-Required-Capability>
```

Figure I-1 illustrates how the Patrol Mission instance is represented in the Protégé ontology editor [3]. In this example, the capability categories have been imported from the Joint C3 Information Exchange Data Model (JC3IEDM) [4], which is an illustration of knowledge reused following Process Activity 3.2.

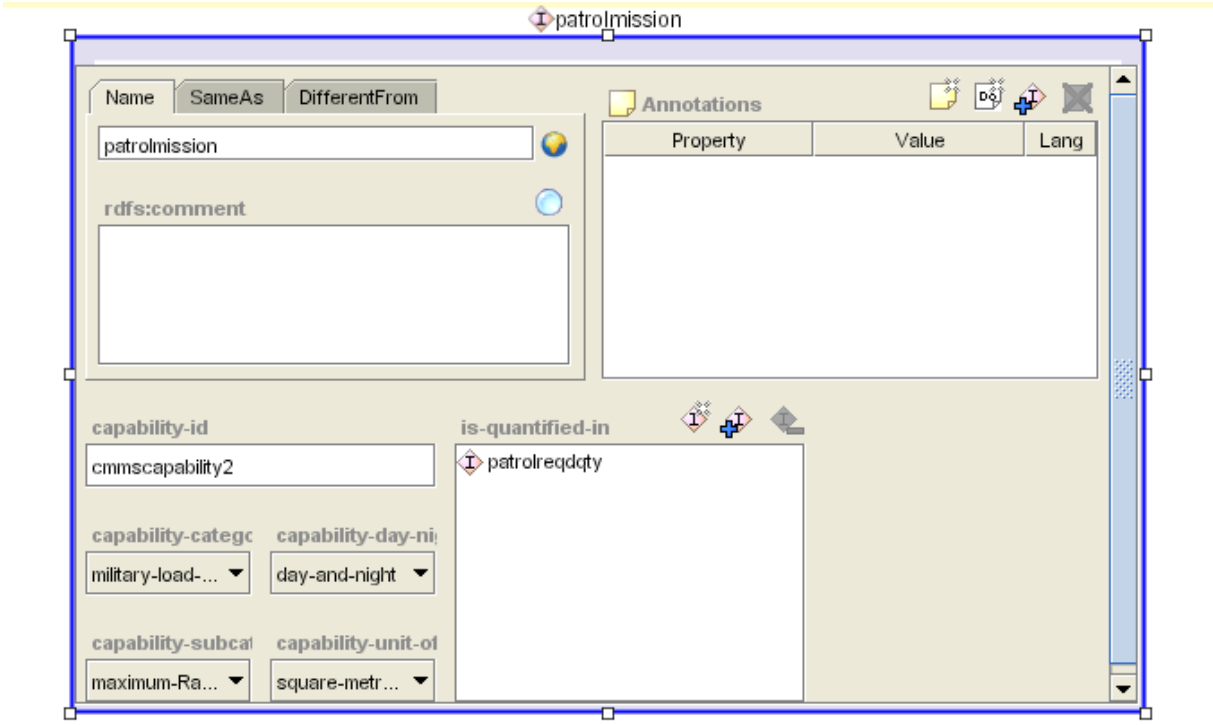

Figure I-1: Sample Scenario Knowledge (Patrol Mission Object) in the Protégé Ontology Editor.

I.2.3 Process Phase 4 – Design Conceptual Model

A conceptual model is used to contain and represent the knowledge in a construct that will fit the type of knowledge acquired during Process Phase 3 and that will allow to make use of that knowledge. In Process Phase 4, the conceptual model design is driven by the intended purpose captured in Product 2.1 – Conceptual Model Requirement Specification. The objective of this example is to illustrate a few design options to fit different usages of the conceptual model. The artefact in Table I-9 is an example of Product 4.1 – Conceptual Model Design.

Table I-9: Conceptual Model Design for the Scenario Description Example.

Component	Design	Build
Conceptual Primitives	*Pool*	*Swedish Contingent, MUST*
	Activity	*Patrol Mission, File Area Clear Report, etc.*
	Event	*Send Search, Interrogate Order/Request*
	Gateway	*Area Secure, Analyze Results*
	Actor	*Intel Officer, Depot Personnel*
	Use Case	*Report File to MUST, Check Information System*
Model Kinds	*Collaboration process diagram, composed of 2 Abstract process diagrams*	
	Use Case	
Views	*Simplest human understandable high-level description of the scenario*	*See Figure I-1*
	Collaboration between organizations	*See Figure I-3*
	Comparison of a scenario procedure to a recommended procedure	*See Figure I-4*
Formalisms	*BPMN*	
	Use Case	
Notations	*BPMN*	
	Custom Pictogram	

Process Activity 4.1 suggests that the conceptual model design may be influenced by the design of another conceptual model being reused. In this example, no existing conceptual model was reused and no constraint of that sort was imposed on the design.

In Process Activity 4.2, conceptual primitives fit for the type of knowledge are selected. The current example involves the conceptual model of a military scenario. Therefore, appropriate conceptual primitives are actors, activities, events, etc. Eligible model kinds that represent interactions between scenario elements include collaboration, activity, sequence, and use case diagrams.

Different formalisms allow representing these combinations of conceptual primitives and model kinds: Petri Nets, ontology, use case, BPMN, etc. In Process Activity 4.3, the formalism choice is influenced by the conceptual model requirement specification. In this example, the conceptual model was intended for visualisation purpose for communication with the subject-matter experts and for process optimization or conformance checking by analysts. Graphical expressiveness was a key driver to present information to participants while robust inference for system interoperability was not an intended use of the conceptual model. The use case formalism was selected for its simplicity (only actors and use cases). BPMN [5] was privileged over Petri Nets and ontology.

The targeted stakeholders and intended purpose drive the selection of views during Process Activity 4.4. Custom views can be created from the generic conceptual model content. For example, a simple high-level description of the scenario is created for communication with the subject-matter experts. An organisational collaboration view and a procedure comparison view are appropriate to the work of the process analysts. In addition to these process views, others views could have useful to other usages, for example component or deployment views to model the communications between information systems.

In Process Activity 4.5, a notation is selected to implement the chosen formalism. BPMN is a formalism that comes with its own notation, a frequent practice in the modeling domain. The BPMN notation was chosen over the UML notation because of its graphical expressiveness, a driving requirement in this example. BPMN supports a few model kinds and their conceptual primitives (Pool, Activity, Event, Gateway). The same Collaboration Process Diagram model kind is used to produce two different views. For the same reason, a custom pictogram notation was selected over the UML notation to express the use case formalism.

Finally, as advised in Process Activity 4.6, the conceptual model conformance to the requirements should be verified. Formal metrics could be derived to measure how fit is the conceptual model for the specific purposes. In this example illustrated in Table I-9, only the overall stakeholders' work efficiency proved the conceptual model usefulness.

I.2.4 Process Phase 5 – Build Conceptual Model

In Process Phase 5, the validated knowledge produced (Product 3.1) is used to populate the content (conceptual primitives) of the conceptual model. The objective of this example is to illustrate the different views generated from the conceptual model based on the knowledge of example I-2.2 and the design choices (formalism, notation, model kinds) summarized in Table I-9.

Figure I-2 presents the simplest human understandable high-level description of the scenario using a custom pictogram notation to express the use case formalism. Figure I-3 shows the collaboration between organizations and Figure I-4 compares the organizational procedures, both using collaboration process diagrams from BPMN.

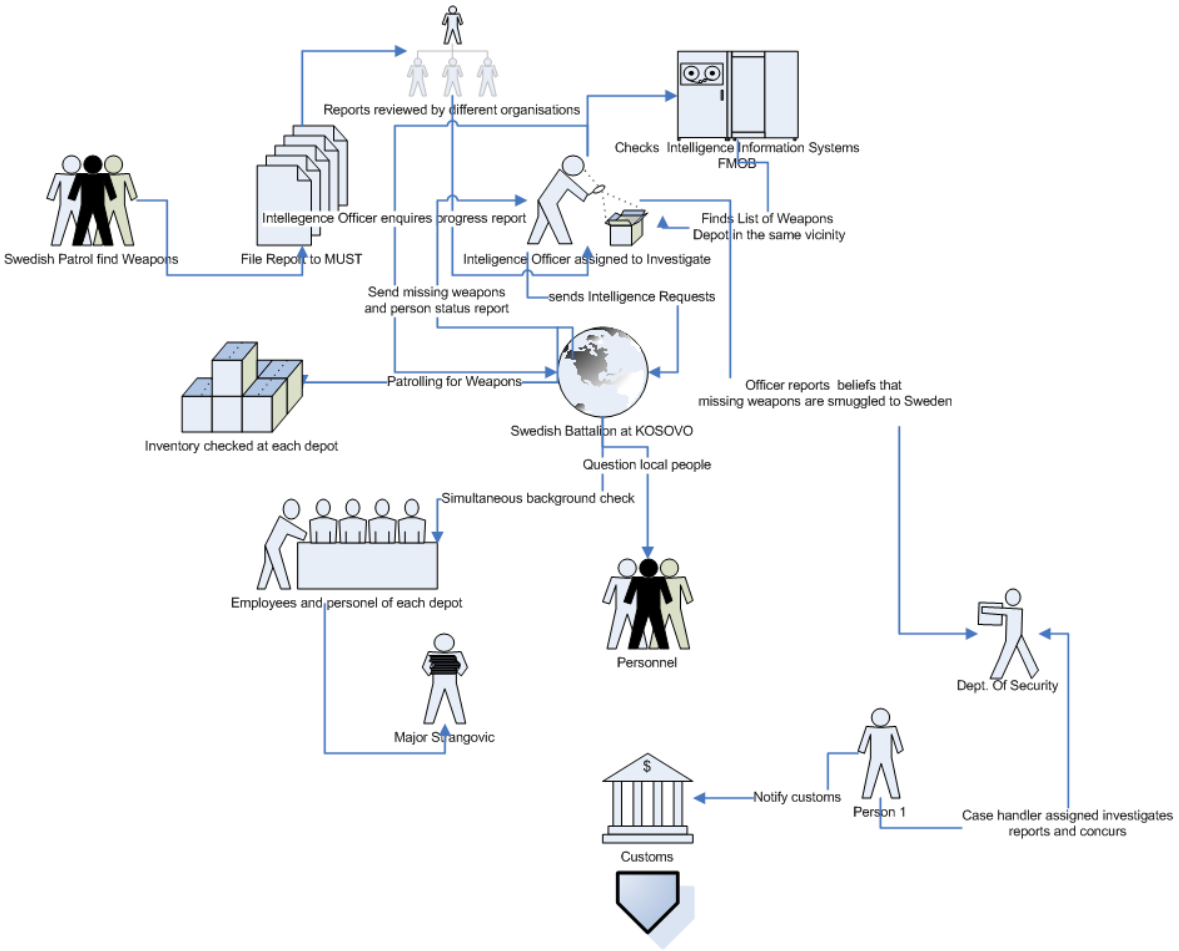

Figure I-2: Sample View of the Conceptual Model of the Scenario Allowing to Visualize the Simplest Human Understandable High-Level Description of the Scenario.

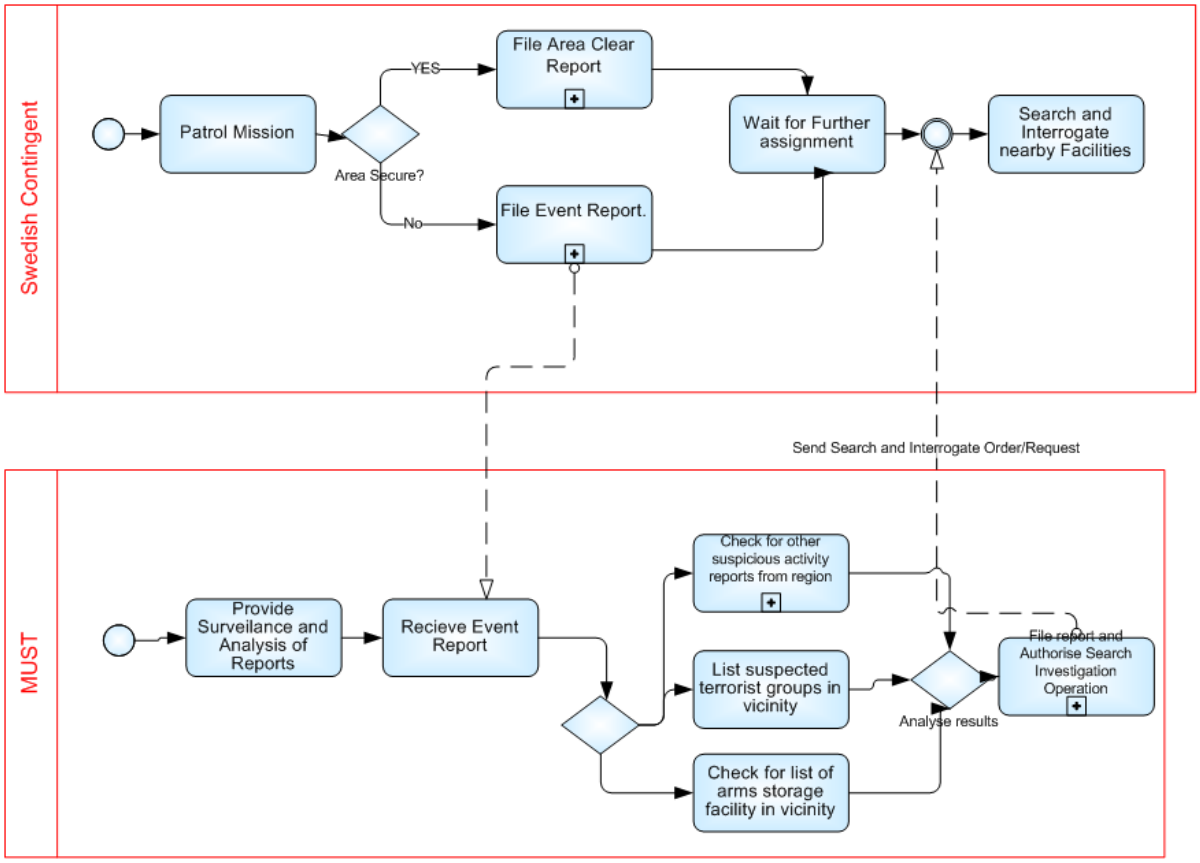

Figure I-3: Sample View of the Conceptual Model of the Scenario Allowing Visualizing the Collaboration Between Organizations.

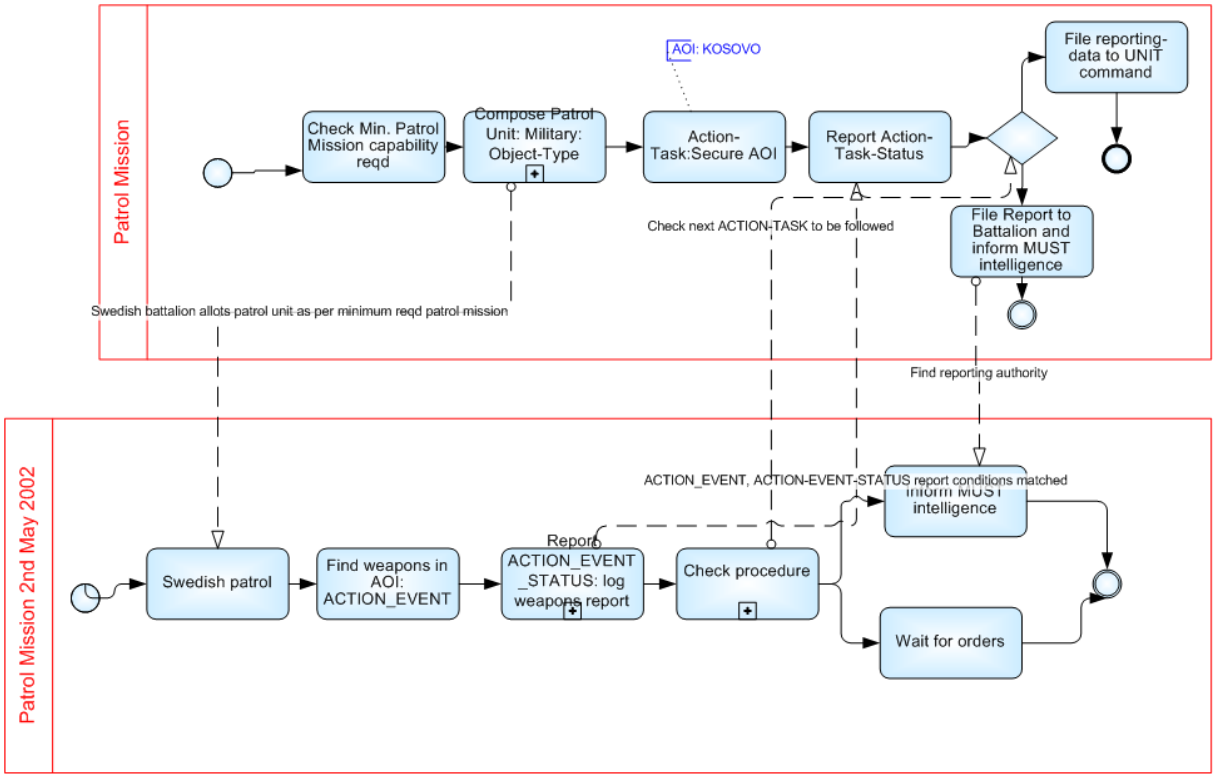

Figure I-4: Sample View of the Conceptual Model of the Scenario Allowing to Compare Procedures.

The artefacts in Figure I-2 to Figure I-4, all together, are an example of Product 5.1 – Conceptual Model.

I.3 RELATION BETWEEN KNOWLEDGE REPRESENTATION AND A CONCEPTUAL MODEL

The proposed conceptual modeling guidance does separate the knowledge documentation in Process Phase 3 from the conceptual model of a simulation in Process Phases 4 and 5. In reality, similar documentation techniques may be used for both. The objective of this example is to differentiate the representation of mission-space knowledge, simulation-space knowledge and the conceptual model of a simulation.

This example is taken from the Canadian IMAGE Project [6]. It presents the conceptual model of a military mission to be simulated. The mission is a humanitarian operation to reconstruct a school in a rugged region akin to Afghanistan. It involves convoys subjected to IEDs and evolving within a dynamical social terrain. Blue friendly forces and red insurgents compete for recruiting the allegiance of the general population. The convoys utilize different roads or tracks over time, some with limiting conditions related to the 3D ruggedness of the terrain. Mine attacks occur with a probability that depends of the type of physical and social grounds being travelled and type of carriers used.

I.3.1 Process Activity 3.4 – Gather, Structure and Document Knowledge

In Figure I-5, the mission-space knowledge is structured and document using conceptual graphs [7] in the CoGUI tool [8]. In Figure I-6, the simulation-space knowledge is represented using a hierarchy of the simulation framework concepts, in this case, a custom Canadian simulation framework.

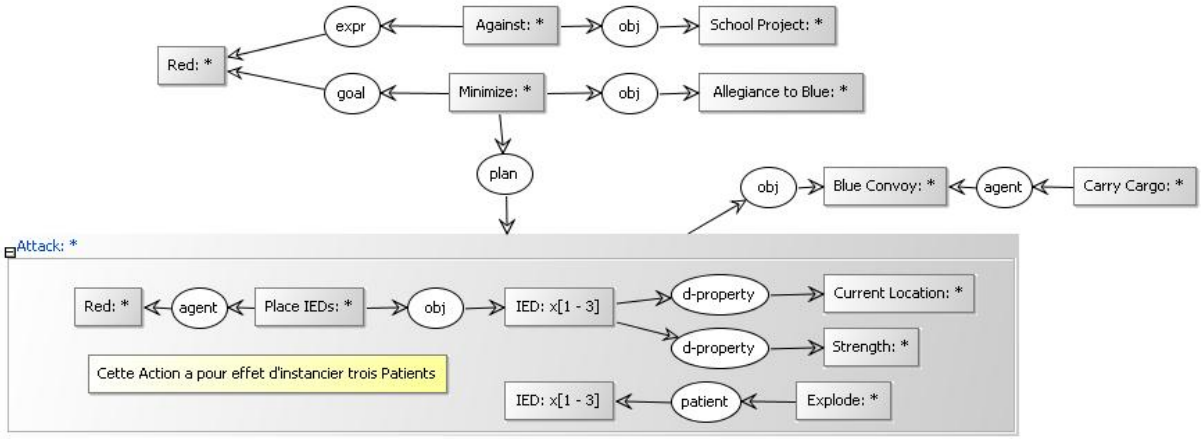

Figure I-5: Sample Mission-Space Knowledge Documentation.

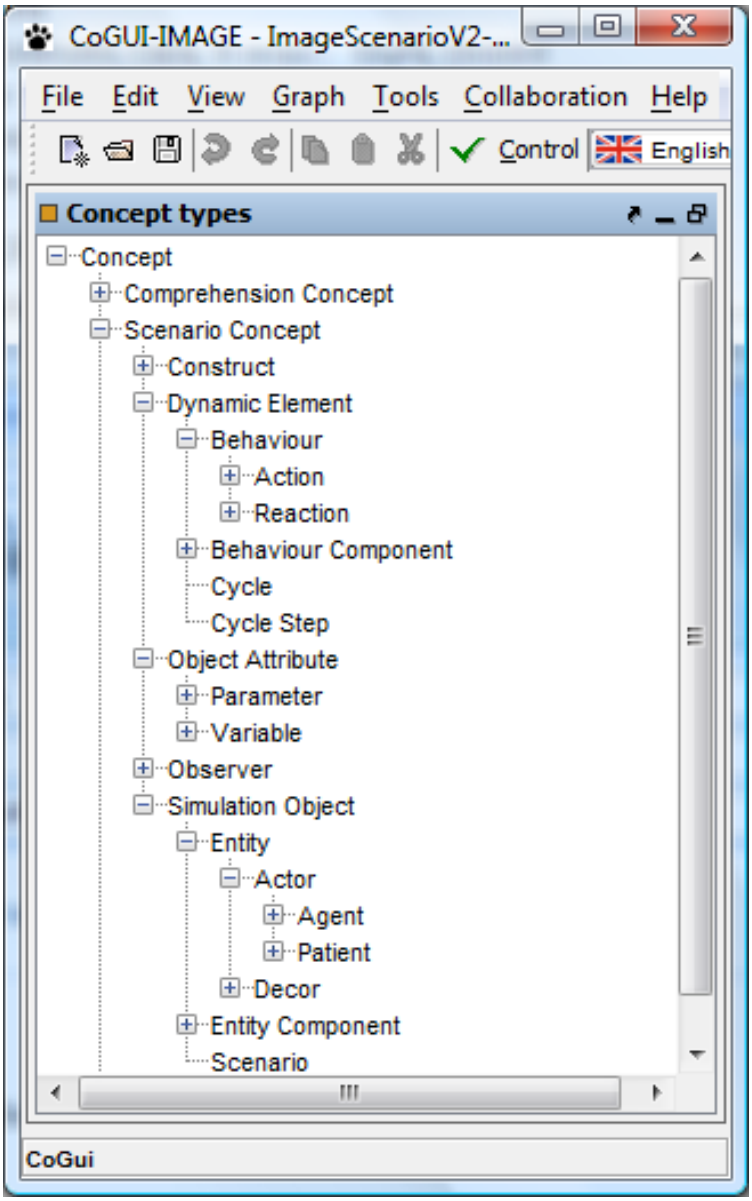

Figure I-6: Sample Simulation-Space Knowledge Documentation.

I.3.2 Process Phase 4 – Design Conceptual Model

In this example, the conceptual graph formalism was pre-selected as an arbitrary choice of the project Group. The conceptual model design components were derived from the formalism. Table I-10 summarizes the conceptual model design components for the conceptual model views of Figure I-5, Figure I-6 and Figure I-7. The artefact in Table I-10 is an example of Product 4.1 – Conceptual Model Design.

Table I-10: Sample Conceptual Model Design for IMAGE.

Component	Design	Build
Conceptual Primitives	*Concept*	*Blue Convoy, Red, IED, etc.*
	Relation	*Agent, Object, Plan, etc.*
	Context	*Attack, Concept Group*
Model Kinds	*Conceptual Graphs*	
Views	*Mission Concepts Inter-Relation*	*See Figure I-5*
	Simulation Concepts Hierarchy	*See Figure I-6*
	Simulation Specification for the Mission	*See Figure I-7*
Formalisms	*Conceptual Graphs*	
Notations	*Conceptual Graphs*	

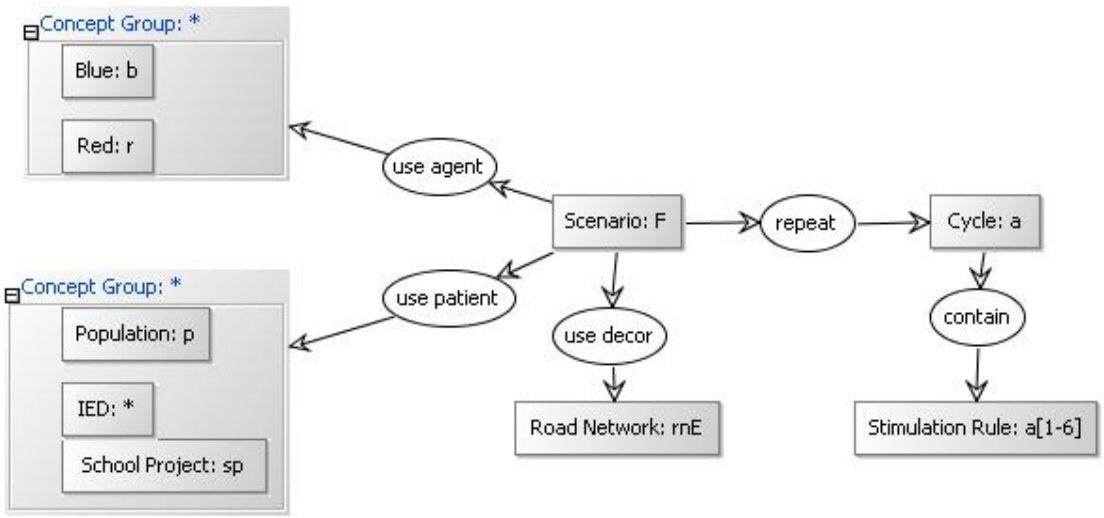

Figure I-7: Sample View from the Conceptual Model of a Simulation.

I.3.3 Process Phase 5 – Build Conceptual Model

Figure I-7 is a conceptual model view illustrating how the scenario concepts are mapped to simulation concepts to describe the specification for the simulation. For example, the Blue and Red concepts become simulation Agents; the Population, IED and School Project become simulation Patients; the Road Network becomes a Decor; and, a simulation Scenario uses the defined Agents, Patients and Decor. Table I-10 includes a mapping of the conceptual primitives and model kinds populated in the conceptual model.

The knowledge representations in Figure I-5 and Figure I-6 can be seen as other views of the conceptual model. This example is representative of the reality where the conceptual model of a simulation often relies on conceptual models of the mission space and the simulation space.

I.4 COMMUNITY-SPECIFIC CONCEPTUAL MODEL DESIGN AND ARTEFACTS

The conceptual modelers have access to a variety of design options. The conceptual model design components being strongly interrelated, one design choice usually imposes constraints on the remaining component options. This explains why different communities have created conceptual model design combinations tailored to their specific domains. The objective of this example is to present sample conceptual model artefacts based on a community-specific conceptual model design.

This example is taken from the United States OneSAF Project [10]. OneSAF Objective System (OOS) is a composable Computer Generated Forces (CGF) simulation framework.

I.4.1 Process Phase 4 – Design Conceptual Model

The OneSAF Group has developed its own conceptual modeling formalism, called CML [11], adapted to model simulations of the tactical military mission space. The CML formalism, notation and conceptual primitives are illustrated in Figure I-8. CML uses a color-coded notation. Table I-11 presents the CML design components.

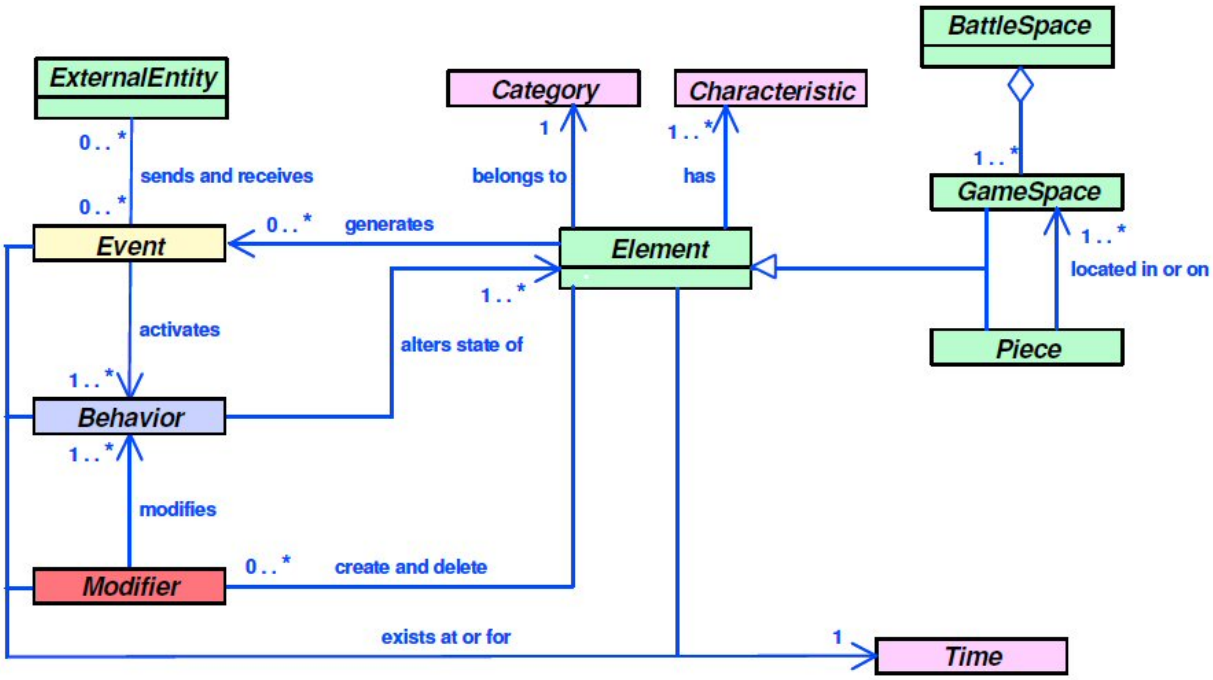

Figure I-8: The CML Formalism, Notation and Conceptual Primitives.

Table I-11: Conceptual Model Design for OneSAF.

Component	Design	Build
Conceptual Primitives	*Element*	*Chassis, Articulation, Weapon, Sensor, Movement Adjudicator*
	Event	*Activate Steering, Fire Command*
	Behavior	*Steering Behavior, Maintain SA, FDC Execute Fire Missions*
	Characteristic	*Movement Freedom, Rotational Freedom*
Model Kinds	*Conceptual Model Language*	
Views	*Entity*	*Entity Composition (see Figure I-9)*
	Common	*Unit Movement Control (see Figure I-10)*
	Battlefield Operating System	*Tactical Fire Direction Center (see Figure I-11)*
Formalisms	*CML*	
Notations	*CML*	

I.4.2 Process Phase 5 – Build Conceptual Model

The OneSAF conceptual model was built according to the CML design. The column "Build" of Table I-11 above includes a few examples of how the components have been populated. Figure I-9 presents a sample Entity view for the Entity Composition. Figure I-10 shows a sample Common view representing the Unit Movement Control. Figure I-11 is an example of a Battlefield Operating System view for the Tactical Fire Direction Center. These artefacts, all together, are part of Product 5.1 – Conceptual Model.

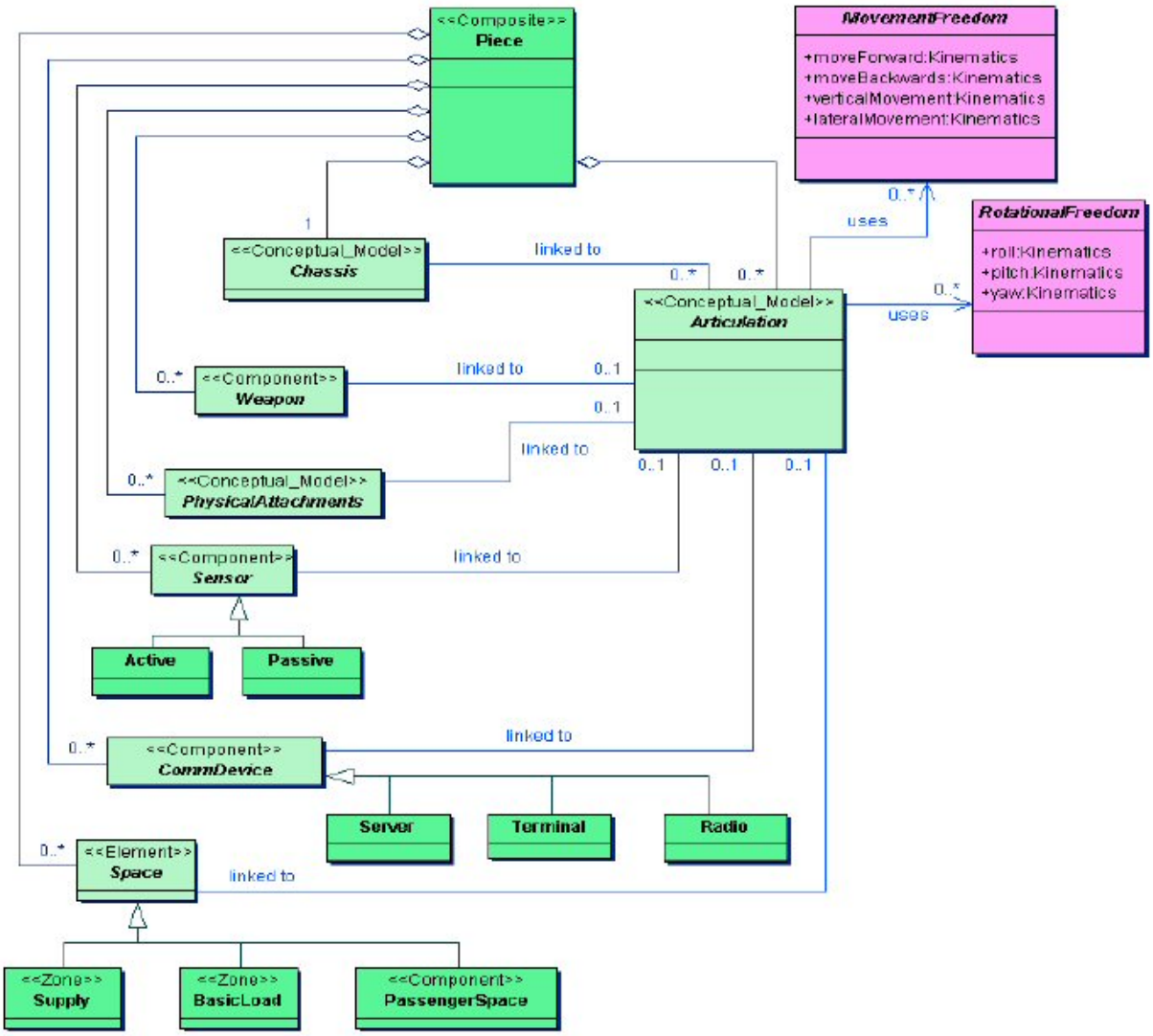

Figure I-9: Sample View from the OneSAF Conceptual Model Representing Entity Composition.

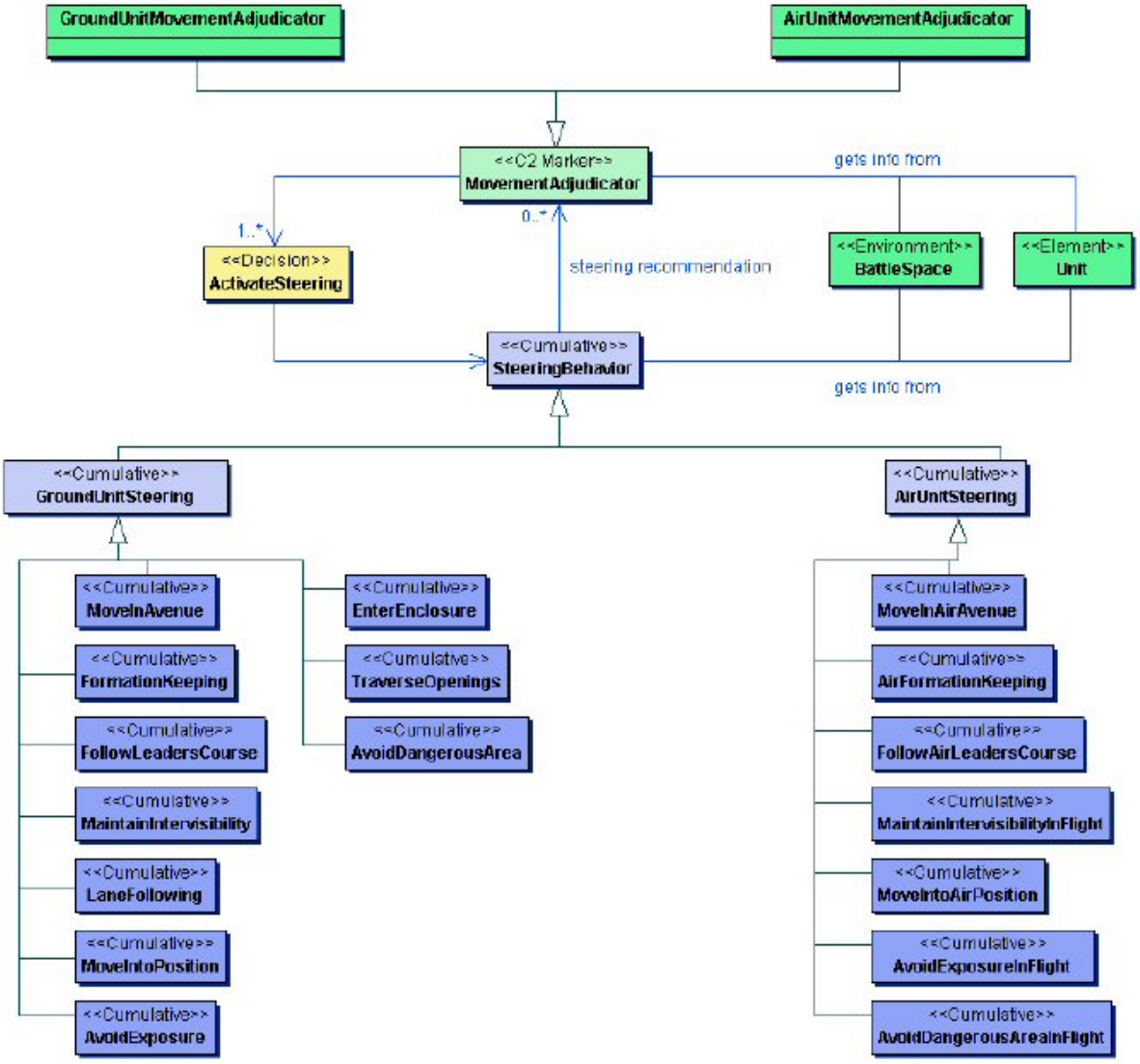

Figure I-10: Sample View from the OneSAF Conceptual Model Representing Unit Movement Control.

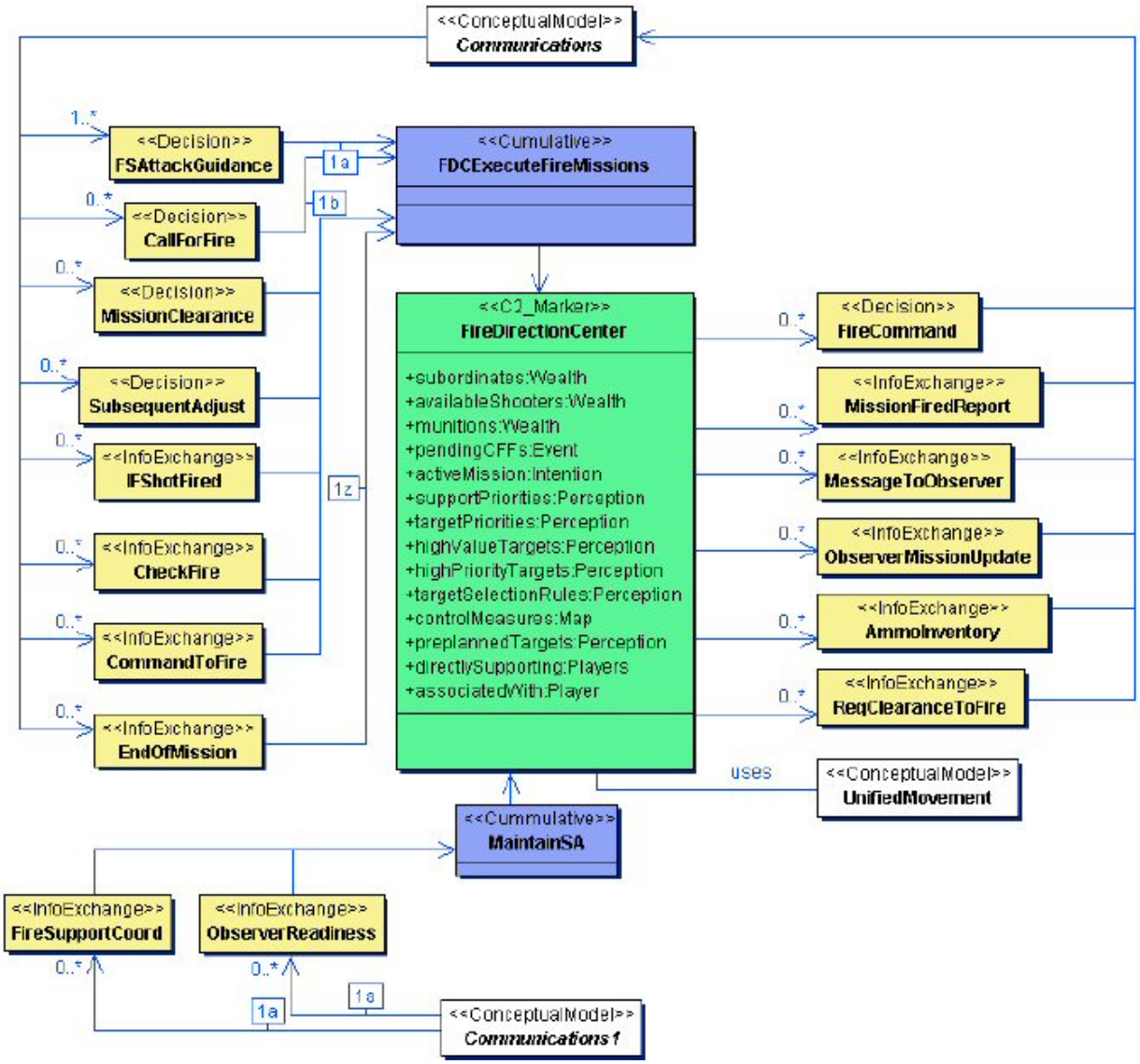

Figure I-11: Sample View from the OneSAF Conceptual Model Representing Tactical Fire Direction Center.

I.5 ITERATIVE EVOLUTION OF CONCEPTUAL MODEL REQUIREMENTS, DESIGN AND ARTEFACTS

Producing preliminary conceptual model artefacts contributes to the learning process and helps refining the conceptual model requirements. The conceptual model design evolves accordingly over several iterations. The objective of this example is to presents the iterative evolution of conceptual model requirements (Process Activity 2.1), design (Process Phase 4) and artefacts (Process Phase 5). New requirements are challenging the conceptual model design and new representation capabilities are incorporated to support the logical thinking process.

This example is taken from the Canadian KARMA Project [11]. KARMA is a simulation framework for guided-weapon engagement simulation intended to be used to develop countermeasures and to assess missile performance. The project started from a blank page, without any legacy simulation system to integrate.

The first conceptual model requirement was to allow the project manager to represent the mission space requirements. The informal view of Figure I-12 was developed using a MindMap design summarized in Table I-12.

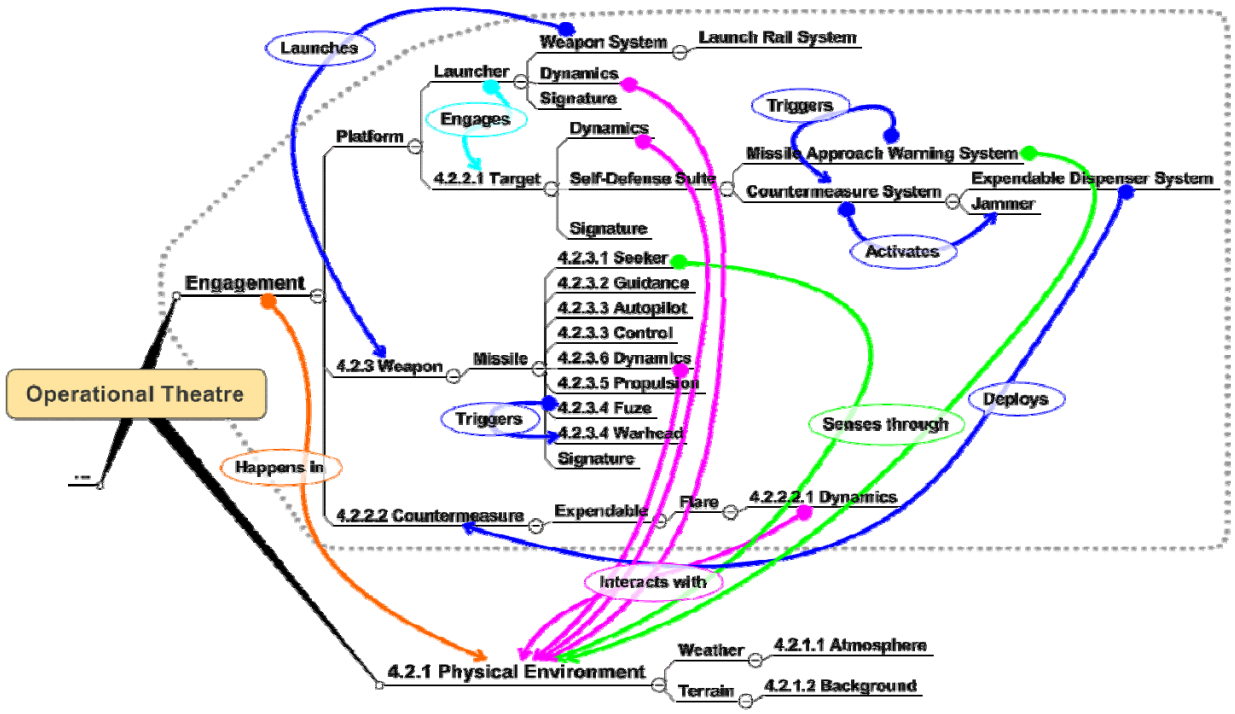

Figure I-12: Sample View of the KARMA Conceptual Model Representing the Engagement Mission Space.

Table I-12: Conceptual Model Design for KARMA Engagement Mission Space.

Component	Design	Build
Conceptual Primitives	*Entity*	*Engagement, Platform, Missile, Autopilot*
	Relationship	*Has a*
	Interaction	*Engages, Triggers, Senses*
Model Kinds	*MindMap*	
Views	*Component*	*Engagement Mission Space (see Figure I-12)*
Formalisms	*MindMap*	
Notations	*Mind Manager® MindMap*	

The second step involved the software architect and the subject-matter experts who needed a conceptual model to support the simulation architecture design. The conceptual model was used to capture the subject-matter experts' knowledge, to engineer the knowledge in order to derive key concepts and to agree on a common understanding to be used as the design reference. Figure I-13 shows the final key concepts diagram. Several iterations of that diagram were produced during working sessions before to adopt a version satisfying the reusability, interoperability and composability requirements. For example, an early version of that diagram included threat/target and red/blue concepts, which was representative of the subject-matter experts' bias. The conceptual model proved to be useful to get rid of that bias. Other views, such as the sequence view in Figure I-14 and the states view in Figure I-15, were used to complete the representation and proof the design. Table I-13 summarizes the conceptual model design components based on the UML notation.

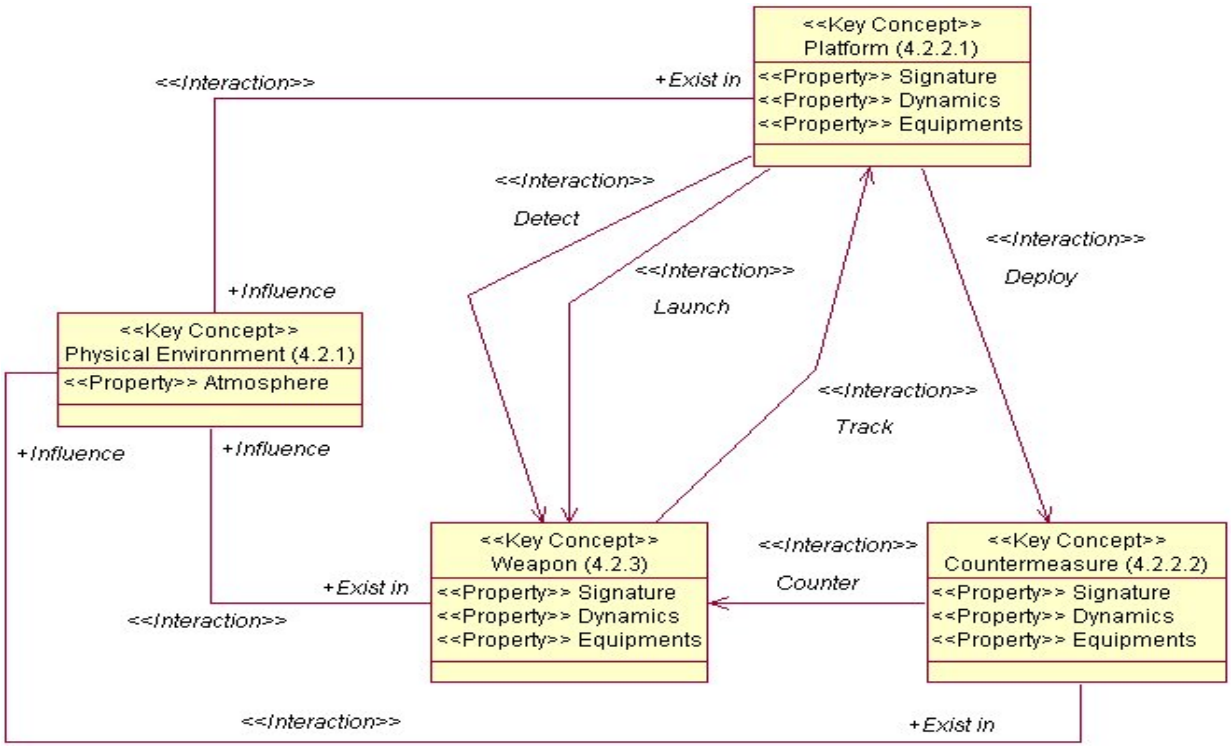

Figure I-13: Sample View of the KARMA Conceptual Model Representing the Engagement Key Concepts.

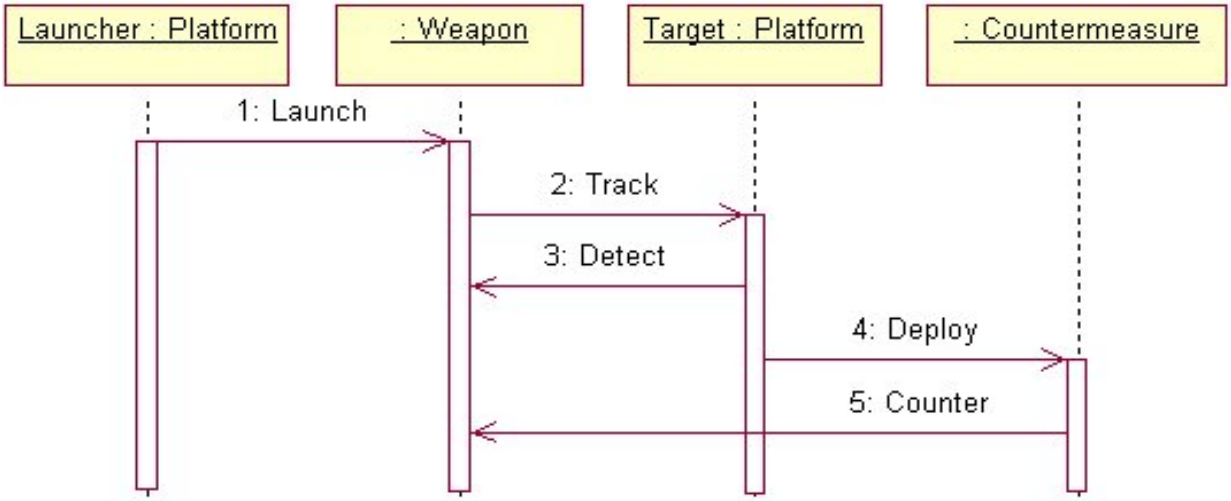

Figure I-14: Sample View of the KARMA Conceptual Model Representing an Engagement Sequence.

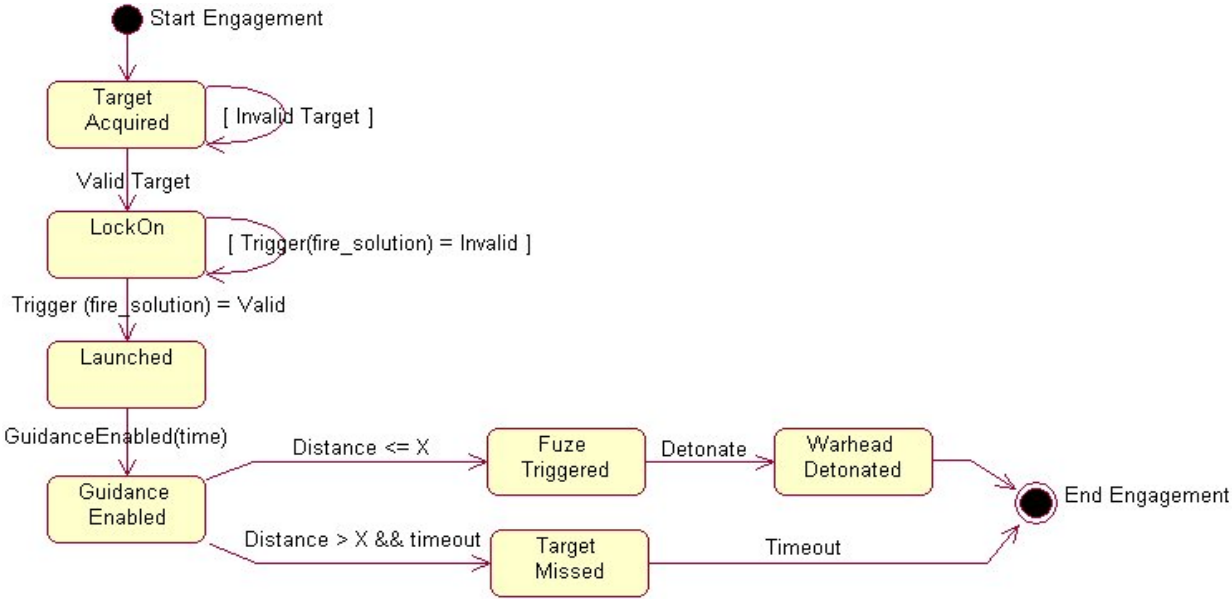

Figure I-15: Sample View of the KARMA Conceptual Model Representing an Engagement States.

Table I-13: Conceptual Model Design for KARMA Engagement Key Concepts.

Component	Design	Build
Conceptual Primitives	*Key Concept (Class)*	*Physical Environment, Platform, Weapon, Countermeasure*
	Property (Attribute)	*Signature, Dynamics, Equipments, Atmosphere*
	Interaction	*Detect, Launch, Track, Deploy, Counter, Exist In, Influence*
	Instance	*Launcher, Target*
	States	*Target Acquired, LockOn, Launched*
Model Kinds	*Class Diagram*	
	Sequence Diagram	
	State Diagram	
Views	*Logicals*	*Engagement Key Concepts (see Figure I-13)*
		Engagement Sequence (see Figure I-14)
		Engagement States (see Figure I-15)
Formalisms	*UML*	
Notations	*UML*	

The conceptual model further evolved to allow the software developers to implement the simulation architecture. The engagement concepts were expressed in the object-oriented formalism as shown in Figure I-16. Simulation space concepts, such as the scheduler and logger mechanisms, were introduced as represented in Figure I-17. Additional primitives (classes, attributes, methods, etc.) completed the model to allow the automatic generation of a C++ implementation from the model.

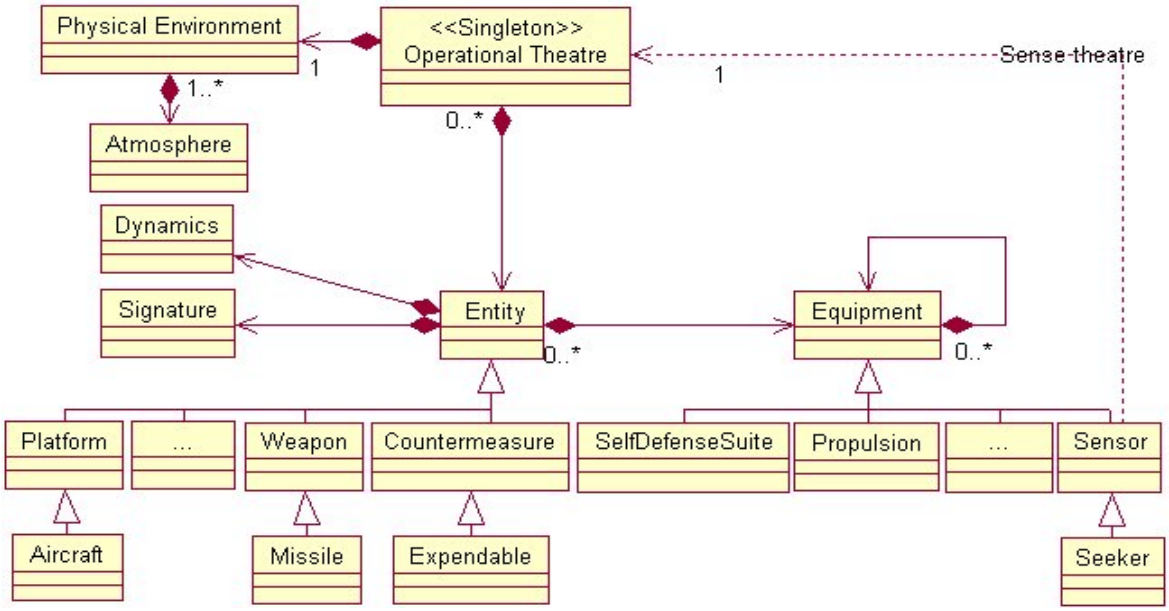

Figure I-16: Sample View of the KARMA Conceptual Model Representing an Engagement in the Object-Oriented Formalism.

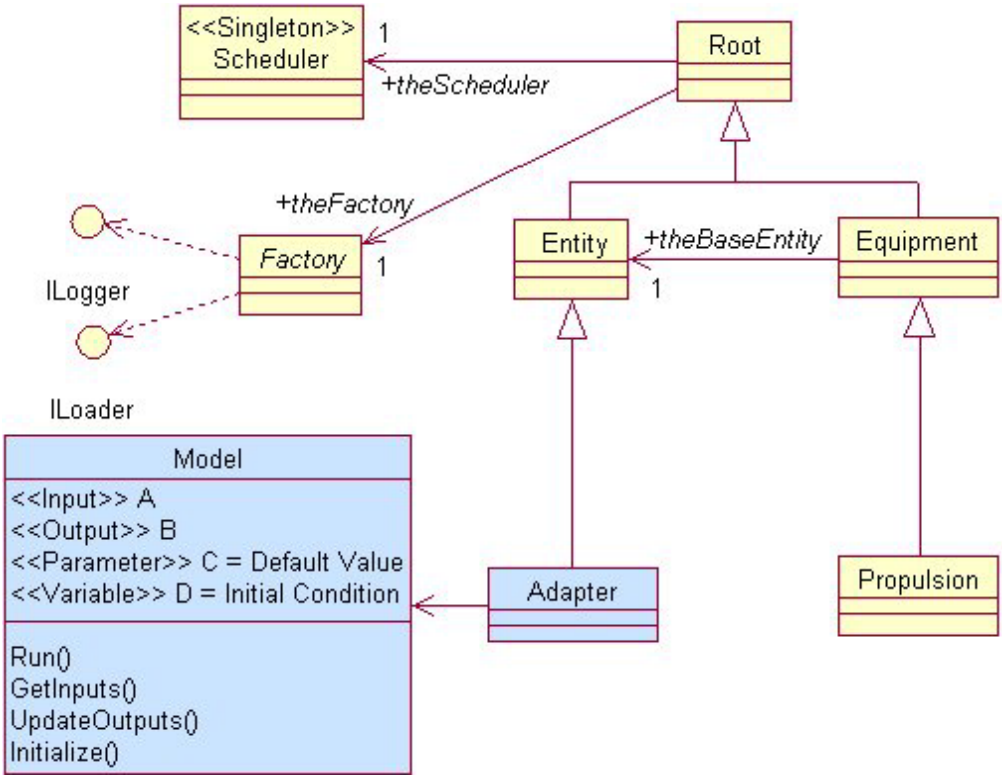

Figure I-17: Sample View of the KARMA Conceptual Model Including Simulation Space Concepts and Implementation-Related Conceptual Primitives.

The complete conceptual allows for the complete traceability from needs to requirements to specification to implementation. Each stakeholder is leveraging the conceptual model for his purpose.

I.6 CONCLUSION

This annex presented a limited number of examples complementing the best-practice guidance. In the interim of a standard conceptual model specification, each enterprise has to specify its own conceptual modeling process and conceptual model products, to a level down to actual templates if required. The proposed guidance should serve as reference and the examples, as inspiration. Every customization of the guidance will contribute to the science of conceptual modeling and will bring valuable experience to the standardization table.

I.7 REFERENCES

[1] Mojtahed, V., Garcia Lozano, M., Svan, P., Andersson, B. and Kabilan, V., DCMF-Defence CM Framework, Systems Technology Methodology Report Number FOI-R–1754–SE, ISSN 1650-1942, FOI-Swedish Defence Research Agency, November 2005, UML: http://www2.foi.se/rapp/foir1754.pdf.

[2] Web Ontology Language (OWL), UML: http://www.w3.org/2004/OWL/.

[3] Protégé ontology editor, URL: http://protege.stanford.edu/.

[4] Joint C3 Information Exchange Data Model (JC3IEDM), URL: http://www.mip-site.org/publicsite/04-Baseline_3.0/JC3IEDM-Joint_C3_Information_Exchange_Data_Model/.

[5] Business Process Modeling Notation (BPMN), URL: http://www.bpmn.org/.

[6] Lizotte, M., Poussart, D., Bernier, F., Mokhtari, M., Boivin, E. and DuCharme, M., IMAGE: Simulation for Understanding Complex Situations and Increasing Future Force Agility, Proceedings of the Army Science Conference 2008, Orlando, FL, USA, December 2008.

[7] Conceptual Graphs, URL: http://conceptualgraphs.org/.

[8] CoGui, URL: http://www.lirmm.fr/cogui/.

[9] OneSAF, URL: http://www.onesaf.net/.

[10] Karr, C.R., CM in OneSAF Objective System, Proceedings of the I/ITSEC 2005 Interservice/Industry Training, Simulation and Education Conference, Paper 2005, Orlando, FL, USA, 28 November – 1 December 2005.

[11] Harrison, N., Gilbert, B., Jeffrey, A., Lestage, R., Lauzon, M. and Morin, A., KARMA: Materializing the Soul of Technologies into Models, Proceedings of the I/ITSEC 2005 Interservice/Industry Training, Simulation and Education Conference, Paper 2256, Orlando, FL, USA, 28 November-1 December 2005.